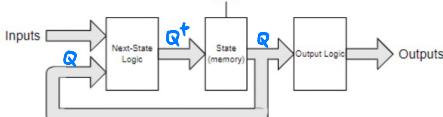


# Analysis of Sequential Circuits

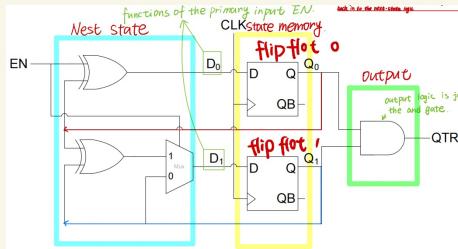
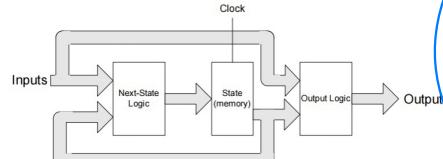
• 时序电路的组成

• 专注于 Synchronous

- ① Moore Output: output = g(current state)



- ② Mealy Output: output = g(current state, inputs)



• 可以写的表达式

Excitation Equations:

$$D_0 = EN \oplus Q_0$$

$$D_1 = EN \cdot (Q_0 + Q_1) + \overline{EN} \cdot Q_1$$

D FF Characteristic Eqn:

$$Q^+ = D$$

Transition Equations:

$$Q_0^+ = D_0 = EN \oplus Q_0$$

$$Q_1^+ = D_1 = EN \cdot (Q_0 + Q_1) + \overline{EN} \cdot Q_1$$

Output Equation:

$$QTR = Q_0 \cdot Q_1$$

已知所要实现的功能,如何设计 Sequential Circuit.

• 所需要的步骤聚.

- (optional) Construct state diagram
- Construct state/output table
- Create state assignments
- Create transition/output table
- Choose FF type
- Construct excitation/output table
  - Similar to transition/output table
- Find excitation and output logic equations

通过 states 的数量判断 state variable 的数量  
 $\lceil \log_2 n \rceil$  (n 是 state 数量)

• 例子 :

Transition/output table

State/Output Table:

S	IN	OUT
zero1s	0 1	0
one1	0 1	0
two1s	0 1	0
three1s	0 1	0
	S*	

该过程就是将 State 用 State Variable 表示

S	Q <sub>1</sub> Q <sub>0</sub>	IN	OUT
zero1s	0 0	0 1	0
one1	0 1	0 1	0
two1s	1 0	0 1	0
three1s	1 1	0 1	0
	S*		

• Choose FF type:

- Using D flip-flops will simplify things (as we'll see below...)
- easiest, because  $Q^+ = D$

• Excitation table

- Shows FF input values required to create next state values for every current state/input combination
- If we're designing with D FFs, entries in excitation/output table are the same as those in transition/output table!

• Excitation Logic

$$D_1 = IN \cdot 0 + IN \cdot (Q_1 \cdot Q_0) \quad D_0 = IN \cdot 0 + IN \cdot (Q_1 \cdot Q_0) = IN \cdot (Q_1 + Q_0)$$

• Output Logic

$$OUT = Q_1 \cdot Q_0$$

$$D_1 = Q_1^+ = \text{function of } (Q_1, Q_0, IN)$$

$$D_0 = Q_0^+ = \text{function of } (Q_1, Q_0, IN)$$

## Implementation Styles

: Random & Regular

### ② Optimization 的标准

: Area & Delay & Testability & Power

### ③ multilevel Synthesis

input 和 output (ff) 之间有

多层次的 logic gates.

④ 用两个 logic gate 的 input literals

的数量来判断 Area 的大小.

因为每个 gate 的每个 input 都

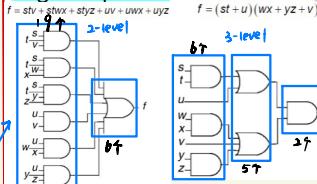
经过 2<sup>t</sup> transistors.

### ⑤ logic expression 和 circuit

是 one-to-one correspondence.

## Logic Synthesis:

### Logic Expressions ⇌ Circuits



### 1.2 level AND and, or gate fan-in # 和什么有关

① 1 level  
# of 1st level AND (OR) gates = # of nontrivial product (sum) terms

### ② 2 level

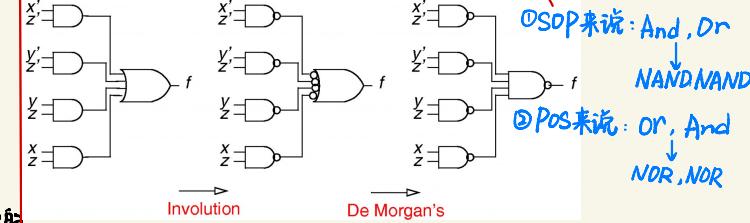
Fan-in of 2nd level OR (AND) gate = total # of product (sum) terms

## 2-Level Circuits

Assumptions:

- Inputs are available in both true and complemented forms (reasonable assumption: FF outputs are available in both phases)
- Gates have no fan-in or fan-out restrictions (unreasonable assumption: typically FI < 4, FO < 6)

• 对于 SOP 和 POS 的 logic gate 进行修改



### Transition/Output Table:

current state	input	output	
		Q <sub>1</sub>	Q <sub>0</sub>
Q <sub>1</sub> Q <sub>0</sub>	0 0	0 1	QTR
0 0	0 0	0 1	0
0 1	0 1	1 0	0
1 0	1 0	1 1	0
1 1	1 1	0 0	1

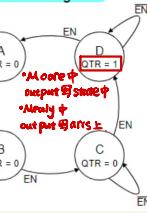
### State/Output Table:

S	EN	QTR
A	0	1
B	1	0
C	0	0
D	1	1

### State labels

Q <sub>1</sub>	Q <sub>0</sub>	State name
0	0	A
0	1	B
1	0	C
1	1	D

### State Diagram:



### ① 对于 Canonical expression (SOP / POS) 进行简化?

### ② Switching algebra theorems

### ③ Karnaugh maps

① 左图的 1 row = 右图的 1 square

② 左图是 normal counting sequence.

右图是 great counting sequence

③ 相邻两个 square 是 adjacent 的关系.

④ 将左图想象成曲面, 四个边两两 adjacent

⑤ 即使重复计算 "1" 的 square, 也优先选择块数多的 minterm combine 方式.

## K map

F	AB	C	00	01	11	10
0	m <sub>0</sub>	m <sub>1</sub>	m <sub>2</sub>	m <sub>3</sub>	m <sub>4</sub>	m <sub>5</sub>
1	m <sub>6</sub>	m <sub>7</sub>	m <sub>8</sub>	m <sub>9</sub>	m <sub>10</sub>	m <sub>11</sub>

F	WX	YZ	XY	ZY
00	00	00	00	00
01	01	01	01	01
11	11	11	11	11
10	10	10	10	10

F	WX	YZ	XY	ZY
00	00	00	00	00
01	11	01	01	01
11	11	11	11	11
10	10	10	10	10

F	WX	YZ	XY	ZY
00	00	00	00	00
01	01	01	01	01
11	11	11	11	11
10	10	10	10	10

Implicant: Any product term that implies a function F (i.e., if, for some input combination, product term P = 1, then F = 1 for the same input combination)

Prime Implicant: An implicant such that if one literal is removed, the resulting product term no longer implies F

Essential Prime Implicant: A prime implicant that covers a minterm that is not covered by any other prime implicants

Using K-maps to find minimal POS

有时 SOP 的 minterm 没有很多能组合则

Step 1:  $F = \text{sop} \rightarrow F = \text{pos}$  想利用 POS.

Step 2: 原来是 0 → 1, 1 → 0, 则得到 F 的表达式.

Step 3: 列出 F 的 SOP →  $F = \overline{sop} = pos$ .

F	WX	YZ	XY	ZY
00	00	00	00	00
01	11	01	01	01
11	11	11	11	11
10	10	10	10	10

F	WX	YZ	XY	ZY
00	00	00	00	00
01	01	01	01	01
11	11	11	11	11
10	10	10	10	10

F	WX	YZ	XY	ZY
00	00	00	00	00
01	01	01	01	01
11	11	11	11	11
10	10	10	10	10

F	WX	YZ	XY	ZY
00	00	00	00	00
01	01	01	01	01
11	11	11	11	11
10	10	10	10	10

F	WX	YZ	XY	ZY
00	00	00	00	00
01	01	01	01	01
11	11	11	11	11
10	10	10	10	10

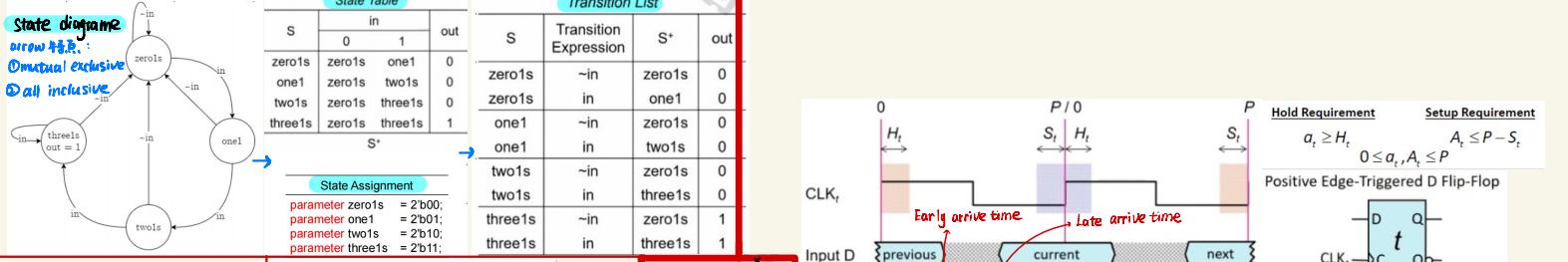
### ④ Excitation equation.

### ⑤ Output equation.

$$D_2 = Q_2^+ = \overline{Q_2} \overline{Q_1} Q_d + \overline{Q_2} Q_1 \overline{Q_0} (n + d)$$

$$+ Q_2 \overline{Q_1} Q_d + Q_2 Q_1 \overline{Q_0} s + Q_2 \over$$

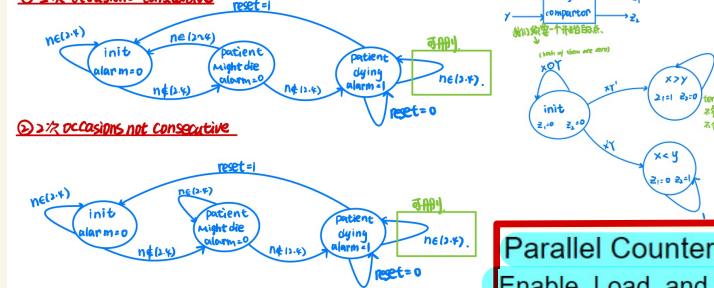
# sequential design example



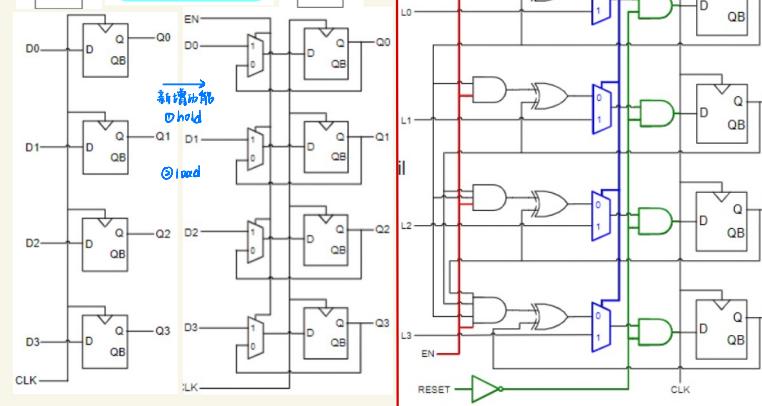
## Sequence Detector V1

```
module SequenceDetectorV1(input clk,
    reg [1:0] Q;
    initial Q = 2'b00;
    // Next-state logic: assign
    wire [1:0] D;
    assign D[1] = in & (Q[1] | Q[0]);
    assign D[0] = in & (Q[1] | ~Q[0]);
    // State Update
    always @(posedge clk) Q <= D;
    // Output logic
    assign out = (Q[1] & Q[0]);
endmodule
```

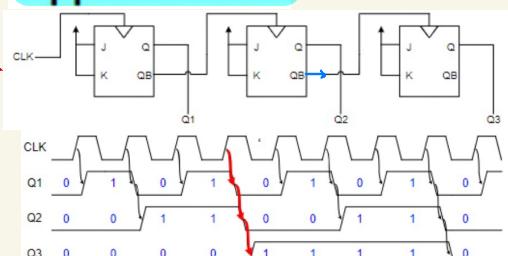
② 2次 occasions consecutive



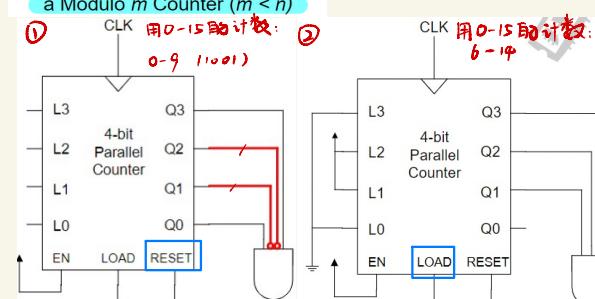
## Registers



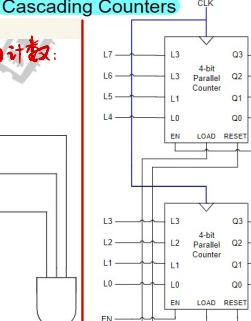
## ripple counter



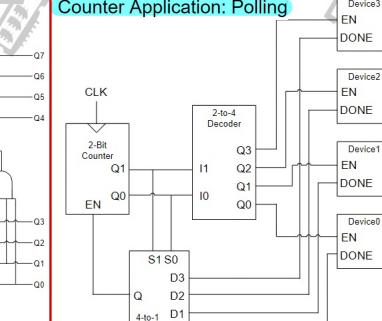
Using a Modulo  $n$  Counter to Make a Modulo  $m$  Counter ( $m < n$ )



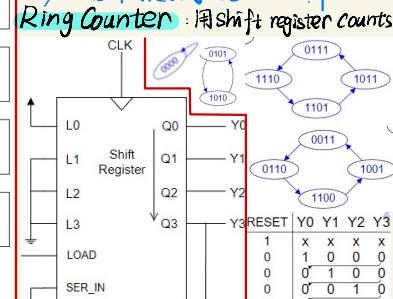
## Cascading Counters



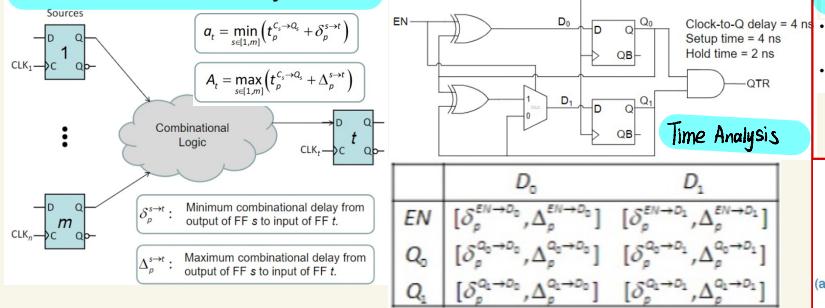
## Counter Application: Polling



Ring Counter: 用 shift register counts  
当后三位不是 0 时 LSB 补上 0. 用 shift register counts



## Combinational Delay Model

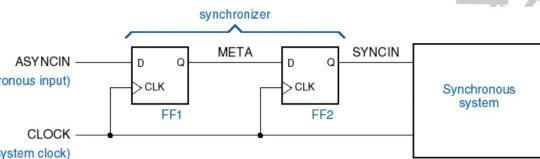


## Initialization

Synchronous vs. Asynchronous Set/Reset

- Synchronous reset: clears Q to 0 on next clock edge
- Synchronous set: sets Q to 1 on next clock edge
- Asynchronous reset: clear Q to 0 immediately (not dependent on clock edge)
  - Example timing diagram shown
- Asynchronous set: set Q to 1 immediately

## Recommended synchronizer

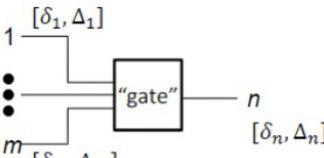


## Path Delay Equations

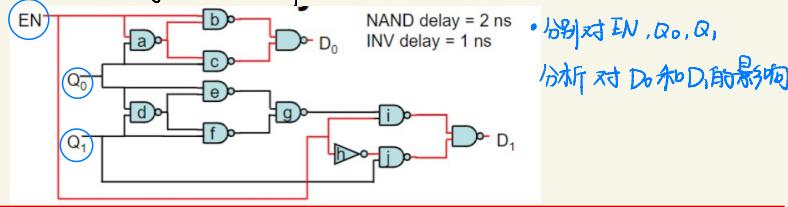
$$[\delta_i, \Delta_i] = \begin{cases} [0, 0] & \text{if } i \text{ is a primary input/FF output} \\ [\min \text{ and max delay to } i] & \text{if } i \text{ is a "gate" output} \end{cases}$$

$$\delta_n = \min_{i \in [1, m]} (\delta_i + t_p^{i \rightarrow n})$$

$$\Delta_n = \max_{i \in [1, m]} (\Delta_i + t_p^{i \rightarrow n})$$



## 分析 Delay 时序分析



## From EN

$$[\delta_p^{EN \rightarrow D_0}, \Delta_p^{EN \rightarrow D_0}] = [4, 6]$$

$$[\delta_p^{EN \rightarrow D_1}, \Delta_p^{EN \rightarrow D_1}] = [4, 5]$$

## From Q0

$$[\delta_p^{Q_0 \rightarrow D_0}, \Delta_p^{Q_0 \rightarrow D_0}] = [4, 6]$$

$$[\delta_p^{Q_0 \rightarrow D_1}, \Delta_p^{Q_0 \rightarrow D_1}] = [8, 10]$$

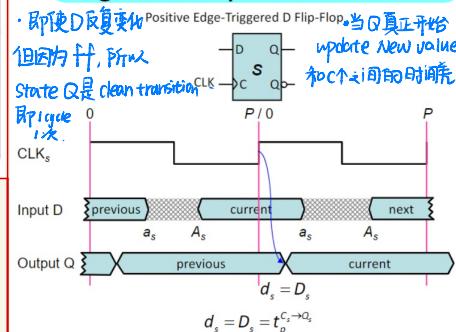
## From Q1

$$[\delta_p^{Q_1 \rightarrow D_0}, \Delta_p^{Q_1 \rightarrow D_0}] = [4, 10]$$

	D0	D1
EN	[4, 6]	[4, 5]
Q0	[4, 6]	[8, 10]
Q1	$\infty, -\infty$	[4, 10]

在算 Timing Analysis 时  
要排除 Asynchronized input : to EN'

## Signal "Departure" Times



$$a_0 = \min(t_p^{C \rightarrow Q} + \delta_p^{Q_0 \rightarrow D_0}, t_p^{C \rightarrow Q} + \delta_p^{Q_1 \rightarrow D_0})$$

$$= \min(4 + 4, 4 + \infty)$$

$$= 8 \text{ ns}$$

$$A_0 = \max(t_p^{C \rightarrow Q} + \Delta_p^{Q_0 \rightarrow D_0}, t_p^{C \rightarrow Q} + \Delta_p^{Q_1 \rightarrow D_0})$$

$$= \max(4 + 6, 4 + \infty)$$

$$= 10 \text{ ns}$$

$$a_1 = \min(t_p^{C \rightarrow Q} + \delta_p^{Q_0 \rightarrow D_1}, t_p^{C \rightarrow Q} + \delta_p^{Q_1 \rightarrow D_1})$$

$$= \min(4 + 8, 4 + 4)$$

$$= 8 \text{ ns}$$

$$A_1 = \max(t_p^{C \rightarrow Q} + \Delta_p^{Q_0 \rightarrow D_1}, t_p^{C \rightarrow Q} + \Delta_p^{Q_1 \rightarrow D_1})$$

$$= \max(4 + 10, 4 + 10)$$

$$= 14 \text{ ns}$$

得到  $A_0, A_1$  后, 就可用于判断最小的 Period & 判断  $a_0, a_1$  是否可以

## Hold Requirement

$$a_0 \geq H$$

$$a_1 \geq H$$

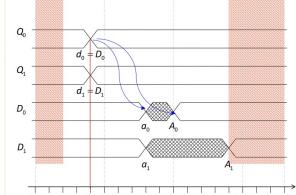
## Setup Requirement

$$A \leq P - S \Rightarrow P \geq A + S$$

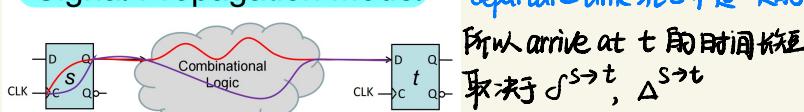
$$P \geq A_0 + S = 10 + 4 = 14 \text{ ns}$$

$$P \geq A_1 + S = 14 + 4 = 18 \text{ ns}$$

$$\therefore P_{\min} = 18 \text{ ns}$$



## Signal Propagation Model



• departure time 在 S 中是一定的  
若从 arrive at t 的时间长  
取决于  $\int s \rightarrow t$ ,  $\Delta s \rightarrow t$

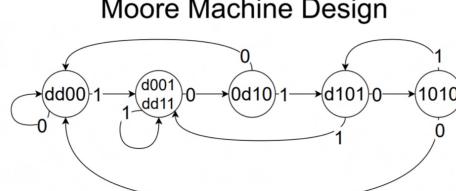
• signal propagation equation

	Shortest path	Longest path
Departure time from s	$d_s = t_p^{C_s \rightarrow Q_s}$	$D_s = t_p^{C_s \rightarrow Q_s}$
Arrival time at t	$a_t = d_s + \delta_p^{s \rightarrow t}$	$A_t = D_s + \Delta_p^{s \rightarrow t}$

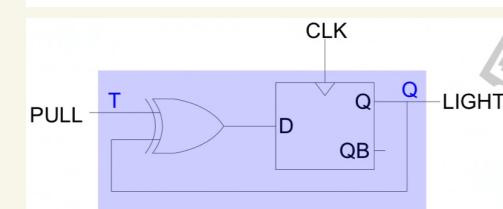
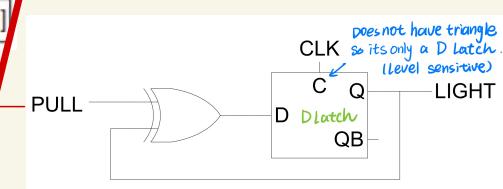
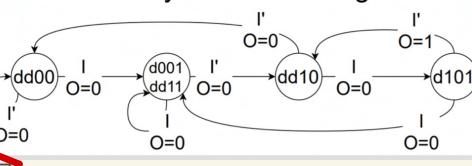
$$a_t = \min_{s \in [1, m]} (t_p^{C_s \rightarrow Q_s} + \delta_p^{s \rightarrow t})$$

$$A_t = \max_{s \in [1, m]} (t_p^{C_s \rightarrow Q_s} + \Delta_p^{s \rightarrow t})$$

## Moore Machine Design



## Mealy Machine Design



This device is known as a toggle flip-flop

sequence	Moore State	Table	I = 0	I = 1	O
"dd00"	A	A	B	0	
"d001" and "dd11"	B	C	B	0	
"0d10"	C	A	D	0	
"d101"	D	E	B	0	
"1010"	E	A	D	1	

sequence	state	Transition	Output O
"dd00"	A	I = 0 A I = 1 B	0 0
"d001" and "dd11"	B	C	0 0
"dd10"	C	A D	0 0
"d101"	D	C B	1 0

